Ø

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/675,817	09/28/2000	Thomas Tomazin	10559-284001 / P9291-ADI	9781

20985 7590 11/28/2005

FISH & RICHARDSON, PC
P.O. BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
HUISMAN, DAVID J	
ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/675,817	Applicant(s) TOMAZIN ET AL.	
	Examiner David J. Huisman	Art Unit 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _30 September 2005_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1,4-8,19-21,24,25 and 27-31_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) _1,4-8,19-21,24-25, and 27_ is/are allowed.

6)☒ Claim(s) _28-31_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _30 September 2005_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 4-8, 19-21, 24-25, and 27-31 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 9/30/2005.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Claim Objections

4. Claim 21 is objected to because of the following informalities: In line 15, replace "of pointer" with --of a pointer--.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 28-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Martin, U.S. Patent No. 4,439,828 (as applied in the previous Office Action and herein referred to as Martin).

7. Referring to claim 28, Martin has taught a method of aligning instructions in a processor comprising:

a) storing a plurality of instructions of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more that sub buffer. See Fig.2 and note buffer areas 18, 19, and 20. Each buffer area includes a plurality of sub-buffers 21, which in turn store portions of instructions having different sizes. See column 3, lines 61-67.

b) aligning a first instruction from said buffer areas. See Fig.2, and note that no matter buffer area(s) the first instruction is stored in, the instruction will be selected, and sent to the instruction register where it will be decoded. See column 4, lines 49-54.

c) decoding the size of the first instruction. See column 4, lines 12-17. Note that the size of the first instruction is determined (decoded) and used by the instruction counter 22.

d) selecting at least one of said plurality of sub-buffers from which to output said first instruction on an output part. See Fig.2, and note that the sub-buffers holding the desired instruction will output the instruction portions on the respective wires, which connect the sub-buffers to the instruction register 24.

e) during said outputting, determining a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction, decoding the size of the second instruction, and determining whether processing the second instruction will deplete at

least one of said plurality of buffer areas. See column 4, lines 12-23. Note that the size of the first instruction is used to increment the instruction counter 22 so that it points to the second instruction. Likewise, when the second instruction is retrieved, its size will be determined so that the instruction counter may locate the third instruction, etc. In addition, it is determined whether the processing of instructions will empty/deplete the buffer.

f) based on said determining whether processing the second instruction will deplete said plurality of buffer areas, instructing the plurality of buffer areas to receive additional instructions. See column 4, lines 17-23, and note that buffer depletion results in replenishing the buffer areas with instructions from main memory.

g) Martin has not explicitly taught determining whether processing the second instruction will deplete at least one of said plurality of buffer areas **based on determining whether the second instruction occupies sub-buffers that include a transition between two of the buffer areas.** However, Martin has taught that each buffer area includes four half-word sub-buffers and that each instruction is one, two, or three half-words in length. See Fig.2 and column 3, lines 61-67. In addition, Martin has taught that the only requirement is that instructions be aligned on half-word boundaries. See column 3, lines 66-67. That is, each instruction may start in any one of the half-word sections of the buffer. Taking an example of an instruction sequence comprising three one-half-word length instructions (instructions A, B, and C) and one three-half-word instruction (instruction D), the first three instructions would take up the first three locations 21 in buffer 18, for instance, thereby leaving one remaining unused location 21 in buffer 18. At this point, there are only two possible solutions to buffering instruction D. It can either be buffered in a fresh, empty buffer, such as buffer 19, so that all of instruction D may be held in one buffer,

or the first half-word of instruction D may be put in the last unused section 21 in buffer 18 and the other two half-words of instruction D may then be put in buffer 19, for instance. A person of ordinary skill in the art would have recognized that the latter is a more ideal and efficient solution because it maximizes memory use. That is, every half-word section in each buffer area would be used. On the other hand, if a new buffer is used every time a partially used buffer has unused sub-buffers but not enough unused sub-buffers to hold a next instruction, then those unused sub-buffers will stay unused. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have instructions transition over multiple buffer areas in order to maximize memory use. And, this is clearly an option because the only requirement set forth by Martin is that instructions are aligned on half-word boundaries (meaning, the can start anywhere in the buffers). It follows that if a second instruction transitions over multiple buffers, then processing that second instruction will deplete portions of multiple buffers, which would result in replenishing those buffers. And clearly, if the instruction takes up two buffers, then the processor will know that when that when that instruction is processed, the two buffers will be able to accept new instructions.

8. Referring to claim 29, Martin has taught a processor comprising:

a) a plurality of buffer areas, each buffer area adapted to store a plurality of instructions of different widths in a plurality of subparts, each of said subparts storing a unit instruction width, and said instruction of greater than a unit instruction width being stored in multiple said subparts. See Fig.2 and note buffer areas 18, 19, and 20. Each buffer area includes a plurality of subparts 21, which in turn store portions of instructions having different widths. See column 3, lines 61-67.

b) a multiplexer, connected to said plurality of subparts, and selecting and aligning at least one of said subparts from any of said subparts within said buffer areas as a current instruction. See Fig.2 and column 4, lines 31-37. Note that instructions are sent to gates 29 and selector 26 chooses which gates to enable such that the associated instructions are outputted. Gates 29 and selector 26 are a multiplexer because a multiplexer selects an item from a plurality of items, and Martin's components perform such a task in that any of the items in the plurality of subparts may be selected for processing.

c) a predictor, operating to predict when at least one of the plurality of buffer areas will be empty, and to send a signal to instruction said at least one of the plurality of buffer areas to load another instruction data. See column 4, lines 12-23, and note that it is determined whether the processing of instructions will empty/deplete the buffer. Note further that buffer depletion results in replenishing the buffer areas with instructions from main memory. The system, upon selecting an instruction each cycle, determines that a portion holding the selected instruction will emptied, and therefore should be refilled.

d) Martin has not explicitly taught operating to predict when at least one of the plurality of buffer areas will be empty **based on determining whether an instruction occupies subparts that include a transition between two of the buffer areas.** However, Martin has taught that each buffer area includes four half-word sub-buffers and that each instruction is one, two, or three half-words in length. See Fig.2 and column 3, lines 61-67. In addition, Martin has taught that the only requirement is that instructions be aligned on half-word boundaries. See column 3, lines 66-67. That is, each instruction may start in any one of the half-word sections of the buffer. Taking an example of an instruction sequence comprising three one-half-word length

instructions (instructions A, B, and C) and one three-half-word instruction (instruction D), the first three instructions would take up the first three locations 21 in buffer 18, for instance, thereby leaving one remaining unused location 21 in buffer 18. At this point, there are only two possible solutions to buffering instruction D. It can either be buffered in a fresh, empty buffer, such as buffer 19, so that all of instruction D may be held in one buffer, or the first half-word of instruction D may be put in the last unused location 21 in buffer 18 and the other two half-words of instruction D may then be put in buffer 19, for instance. A person of ordinary skill in the art would have recognized that the latter is a more ideal and efficient solution because it maximizes memory use. That is, every half-word section in each buffer area would be used. On the other hand, if a new buffer is used every time a partially used buffer has unused sub-buffers but not enough unused sub-buffers to hold a next instruction, then those unused sub-buffers will stay unused. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have instructions transition over multiple buffer areas in order to maximize memory use. And, this is clearly an option because the only requirement set forth by Martin is that instructions are aligned on half-word boundaries (meaning, the can start anywhere in the buffers). It follows that if a second instruction transitions over multiple buffers, then processing that second instruction will deplete portions of multiple buffers, which would result in replenishing those buffers. And clearly, if the instruction takes up two buffers, then the processor will know that when that when that instruction is processed, the two buffers will be able to accept new instructions.

9. Referring to claim 30, Martin has taught an apparatus, including instructions residing on a machine-readable storage medium, for use in a machine system to align instructions in a processor, the instructions causing the machine to:

a) store a plurality of instruction of different sizes in a plurality of buffer areas, each buffer area including a plurality of sub-buffers, each sub-buffer storing a unit instruction width, with an instruction of greater than a unit instruction width stored in more than one sub-buffer. See Fig.2 and note buffer areas 18, 19, and 20. Each buffer area includes a plurality of sub-buffers 21, which in turn store portions of instructions having different sizes. See column 3, lines 61-67.

b) decode a size of a first instruction from said plurality of buffer areas. See column 4, lines 12-17. Note that the size of the first instruction is determined (decoded) and used by the instruction counter 22.

c) select at least one of said plurality of sub-buffers from which to output said first instruction on an output part. See Fig.2, and note that the sub-buffers holding the desired instruction will output the instruction portions on the respective wires, which connect the sub-buffers to the instruction register 24.

d) during said outputting, determine a beginning of a second instruction from selected ones of the plurality of sub-buffers based on the size of the first instruction, decode the size of the second instruction, and determine whether processing the second instruction will deplete at least one of said plurality of buffer areas. See column 4, lines 12-23. Note that the size of the first instruction is used to increment the instruction counter 22 so that it points to the second instruction. Likewise, when the second instruction is retrieved, its size will be determined so that

the instruction counter may locate the third instruction, etc. In addition, it is determined whether the processing of instructions will empty/deplete the buffer.

e) based on said determining whether processing the second instruction will deplete said plurality of buffer areas, instruct the plurality of buffer areas to receive additional instructions. See column 4, lines 17-23, and note that buffer depletion results in replenishing the buffer areas with instructions from main memory.

f) Martin has not explicitly taught determining whether processing the second instruction will deplete at least one of said plurality of buffer areas **based on determining whether the second instruction occupies sub-buffers that include a transition between two of the buffer areas.** However, Martin has taught that each buffer area includes four half-word sub-buffers and that each instruction is one, two, or three half-words in length. See Fig.2 and column 3, lines 61-67. In addition, Martin has taught that the only requirement is that instructions be aligned on half-word boundaries. See column 3, lines 66-67. That is, each instruction may start in any one of the half-word sections of the buffer. Taking an example of an instruction sequence comprising three one-half-word length instructions (instructions A, B, and C) and one three-half-word instruction (instruction D), the first three instructions would take up the first three locations 21 in buffer 18, for instance, thereby leaving one remaining unused location 21 in buffer 18. At this point, there are only two possible solutions to buffering instruction D. It can either be buffered in a fresh, empty buffer, such as buffer 19, so that all of instruction D may be held in one buffer, or the first half-word of instruction D may be put in the last unused section 21 in buffer 18 and the other two half-words of instruction D may then be put in buffer 19, for instance. A person of ordinary skill in the art would have recognized that the latter is a more ideal and efficient

solution because it maximizes memory use. That is, every half-word section in each buffer area would be used. On the other hand, if a new buffer is used every time a partially used buffer has unused sub-buffers but not enough unused sub-buffers to hold a next instruction, then those unused sub-buffers will stay unused. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have instructions transition over multiple buffer areas in order to maximize memory use. And, this is clearly an option because the only requirement set forth by Martin is that instructions are aligned on half-word boundaries (meaning, the can start anywhere in the buffers). It follows that if a second instruction transitions over multiple buffers, then processing that second instruction will deplete portions of multiple buffers, which would result in replenishing those buffers. And clearly, if the instruction takes up two buffers, then the processor will know that when that when that instruction is processed, the two buffers will be able to accept new instructions.

10. Referring to claim 31, Martin has taught a method of processing instructions within a processor, comprising:

a) storing instructions of different widths within a cache having a plurality of buffer areas, each buffer area having a plurality of subportions, each subportion in the cache storing a unit instruction width, where an instruction of unit width takes up a single subportion in the cache, and an instruction of more than said unit width takes up more than one subportion within the cache. See Fig.2 and note buffer areas 18, 19, and 20. Each buffer area includes a plurality of subportions 21, which in turn store portions of instructions having different widths. See column 3, lines 61-67.

b) multiplexing each of the subportions of said cache to an output point, and selecting contents of at least one of said cache subportions as a current instruction. See Fig.2 and column 4, lines 31-37. Note that instructions are sent to gates 29 and selector 26 chooses which gates to enable such that the associated instructions are outputted. Gates 29 and selector 26 perform multiplexing because multiplexing is the selection of an item from a plurality of items and any of the instructions in the plurality of subportions may be selected.

c) during said selecting said current instruction, predicting which of said buffer areas within said cache will be depleted of instruction data, and instructing loading of that number of buffer areas with additional instruction information. See column 4, lines 12-23, and note that it is determined whether the processing of instructions will empty/deplete the buffer. Note further that buffer depletion results in replenishing the buffer areas with instructions from main memory. The system, upon selecting an instruction each cycle, determines that a portion holding the selected instruction will emptied, and therefore should be refilled.

d) Martin has not explicitly taught predicting which of said buffer areas within said cache will be depleted of instruction data **based on determining whether an instruction occupies subportions that include a transition between two of the buffer areas.** However, Martin has taught that each buffer area includes four half-word sub-buffers and that each instruction is one, two, or three half-words in length. See Fig.2 and column 3, lines 61-67. In addition, Martin has taught that the only requirement is that instructions be aligned on half-word boundaries. See column 3, lines 66-67. That is, each instruction may start in any one of the half-word sections of the buffer. Taking an example of an instruction sequence comprising three one-half-word length instructions (instructions A, B, and C) and one three-half-word instruction (instruction D), the

first three instructions would take up the first three locations 21 in buffer 18, for instance, thereby leaving one remaining unused location 21 in buffer 18. At this point, there are only two possible solutions to buffering instruction D. It can either be buffered in a fresh, empty buffer, such as buffer 19, so that all of instruction D may be held in one buffer, or the first half-word of instruction D may be put in the last unused section 21 in buffer 18 and the other two half-words of instruction D may then be put in buffer 19, for instance. A person of ordinary skill in the art would have recognized that the latter is a more ideal and efficient solution because it maximizes memory use. That is, every half-word section in each buffer area would be used. On the other hand, if a new buffer is used every time a partially used buffer has unused sub-buffers but not enough unused sub-buffers to hold a next instruction, then those unused sub-buffers will stay unused. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have instructions transition over multiple buffer areas in order to maximize memory use. And, this is clearly an option because the only requirement set forth by Martin is that instructions are aligned on half-word boundaries (meaning, the can start anywhere in the buffers). It follows that if a second instruction transitions over multiple buffers, then processing that second instruction will deplete portions of multiple buffers, which would result in replenishing those buffers. And clearly, if the instruction takes up two buffers, then the processor will know that when that when that instruction is processed, the two buffers will be able to accept new instructions.

Allowable Subject Matter

11. Claims 1, 4-8, 19-21, 24-25, and 27 are allowed.

Conclusion

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
November 21, 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100